# 《Java 特性篇》

## java语言有什么特性，继承有什么用处，多态有什么用处

答：java有八大特性：

1.1 面向对象语言，具备抽象、封装、继承、多态特性。

1.2 平台无关性，jvm屏蔽了操作系统差异，使得java语言通过字节码在不同操作系统的jvm上被差异化解释，实现java语言的平台无关性。

1.3 解释型语言，不同于C++、C语言，编译生成针对CPU的机器码，java语言编译生成字节码，会按句翻译执行。在jit即使编译技术加持下，可以将字节码转化成高效的本地机器码，提高执行效率。

1.4 多线程，java内置多线程支持，比C++语言更加方便，不需要调用操作系统多线程功能接口即可完成程序设计。

1.5 安全，java小程序运行在java环境中，不允许它访问计算机的其他部分。

1.6 动态编译，由于java类来自于开发者编写、类库引入，类是动态加载的，可以在分布式环境中动态地维护程序和类库，不用像C++，每次类库升级需要重新编译整个程序。

继承的作用：很方便地复用代码+不破坏现有代码的结构和功能。多态的用处：分离做什么和怎么做，进一步分离接口和实现。不仅可以改善代码组织结构和可读性，而且可扩展性也很强。

## 反射是什么，在哪里用到，怎么利用反射创建一个对象

在运行期间打开和检查java类，并提供获取方法、构造器和字段的接口的类库，在跨网络远程调用、基于RAD可视化编程中用到；

根据class名称获取类字节码对象，通过newInstance()调用无参构造器；getConstructor获取有参构造器，并完成对象构造。

2.1 Class<?> cls = Class.forName("class.name"); Object obj = cls.newInstance();

2.2 Class<?> cls = Class.forName("class.name"); Construct<?> con = cls.getConstructor(String.class,String.class); Object obj = con.newInstance(str1,str2);

## 对象加载的过程，属性先加载还是方法先加载

父类静态属性 (可以是对象) 和静态代码块，看其在类中的先后顺序

子类静态属性和静态代码块 ，看其在类中的先后顺序

父类非静态属性和非静态代码块 ，看其在类中的先后顺序

父类构造方法

子类非静态属性和非静态代码块 ，看其在类中的先后顺序

子类构造方法

规律：

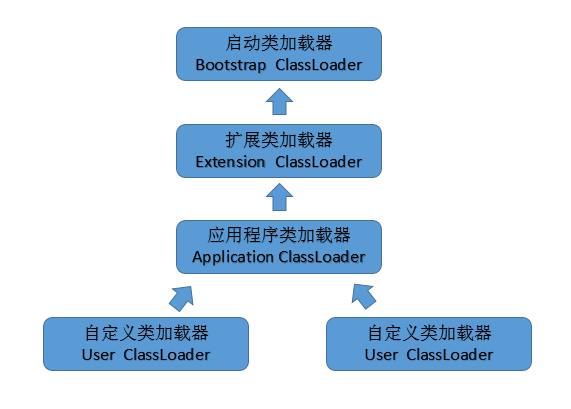
1.静态属性和代码块，跨类高优先级；

2.非静态属性和代码块同类中优先级高于构造函数；

3.静态和非静态的属性和代码块的加载顺序，同对应的定义顺序。

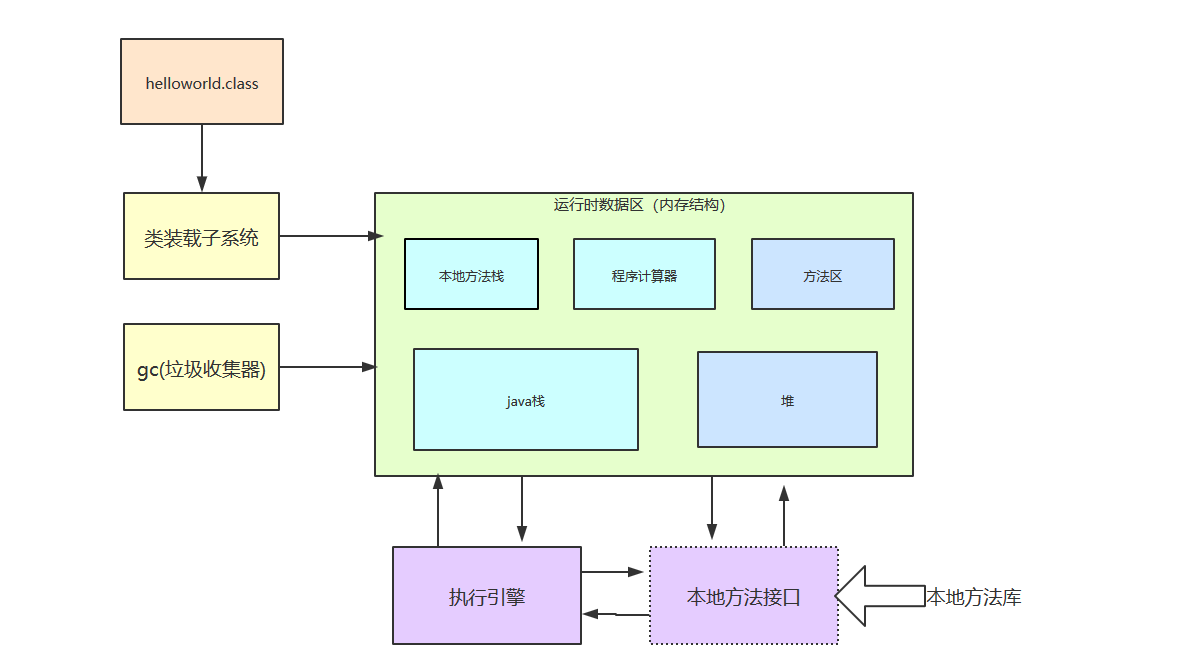
## 垃圾回收机制与jvm结构

**类加载：**java虚拟机是由启动类加载器加载的，负责执行java程序；java类加载器的结构：启动类加载器-->扩展类加载器-->应用程序类加载器-->用户自动定义加载器，满足双亲委派原则，保护java核心库类不被轻易篡改、避免类重复加载。



java类实例相同的判断：1.相同的类路径 2.同一个类加载器(避免不同虚拟机加载同一个class文件的情况)

**内存结构：**



《设计模式》

1.代理模式与装饰模式的区别，手写一个静态代理，一个动态代理

A.代理模式，主要用于为跨进程访问、安全控制、开销很大的对象提供访问层。++

B.装饰模式，用于在不破坏现有类结构的前提下，对类对象就行功能追加，相对继承更加灵活、方便。

C.静态代理, 代理类和目标类相对简单单一，代理类对目标类的访问控制、执行控制等进行增强。

D.动态代理，java运行期间，根据代码业务需要，通过JDK或CGLIB提供的接口(反射机制)，为指定的目标对象，生成对应的代理对象。

代码见Examples-->DynamicProxy

6.自定义View,事件分发机制讲一讲

7.http与https有什么区别

8.Activity启动模式，以及各启动模式生命周期问题

9.静态方法，静态对象为什么不能继承

10.Activity怎么启动Service，Activity与Service交互，Service与Thread的区别

11.介绍一下android动画

12.Launcher启动App的流程，中间有几种跨进程通信(socket)

13.Handler通信，Binder通信

14.你碰到过什么内存泄漏，怎么处理

15.RXJava怎么切换线程

16.Fragment hide show生命周期

17.平常有用到什么锁，synchronized底层原理是什么